UDA-SVD: UPSTREAM- AND DOWNSTREAM-AWARE LOW-RANK APPROXIMATION AND ADAPTIVE RANK ALLOCATION FOR LLM COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid growth of large language models (LLMs) has intensified the demand for efficient compression, and low-rank factorization provides a simple and broadly compatible solution. However, in sequential low-rank compression, the effectiveness of compressing each module is shaped by two factors: ❶ upstream compression perturbs the activations received by the current module; and ❷ approximation errors introduced at the current module propagate downstream and can have unequal effects on the final prediction. Existing methods typically model these effects in isolation. Conventional layer-wise approaches account for neither explicitly, whereas cumulative error-compensation methods primarily address upstream deviations and loss-aware methods focus on downstream sensitivity. Moreover, uniform rank allocation ignores differences in rank sensitivity across modules, leading to inefficient capacity allocation. We propose UDA-SVD, an upstream- and downstream-aware framework for low-rank LLM compression. UDA-SVD captures both activation deviations accumulated through upstream compression and the sensitivity of the language-modeling objective to compression errors at the current module. By integrating these two signals into a unified objective, UDA-SVD provides a common optimization basis for module-wise low-rank approximation and adaptive rank allocation under a fixed parameter budget. Across multiple compression ratios and model architectures, UDA-SVD consistently achieves the highest downstream accuracy among the compared SVD-based methods while substantially reducing language-modeling perplexity, including a nearly 40% reduction on WikiText-2. These results demonstrate the effectiveness of combining upstream drift, downstream sensitivity, and adaptive rank allocation, highlighting UDA-SVD as a promising framework for effective low-rank LLM compression.

1 INTRODUCTION

Large language models (LLMs), exemplified by the GPT (Achiam et al., 2023; Dettmers et al., 2022), LLaMA (Touvron et al., 2023b; Grattafiori et al., 2024), and Qwen (Bai et al., 2023; Team, 2024; Yang et al., 2025) families, have achieved remarkable performance across a wide range of language understanding, reasoning, and generation tasks. However, their large parameter counts incur substantial memory and computational costs, limiting deployment in resource-constrained environments. Model compression provides an effective means of alleviating these costs, among which low-rank factorization (Yuan et al., 2023; Wang et al., 2024; Li et al., 2025) has emerged as a broadly applicable approach. Typically based on truncated singular value decomposition (SVD) (Eckart & Young, 1936), low-rank compression factorizes dense weight matrices into compact low-rank components, thereby reducing both parameter storage and arithmetic cost. Its hardware-agnostic implementation requires no specialized sparse kernels and remains compatible with complementary compression techniques, making it particularly attractive for efficient LLM deployment.

Post-training SVD compression is typically applied sequentially, so the effectiveness of compressing a module depends on both preceding and subsequent computations. Replacing an upstream module with low-rank factors perturbs the activations received by later modules, causing them to deviate from their full-precision counterparts. Meanwhile, errors introduced when compressing the current module propagate through the remaining network and can affect the final language-modeling ob-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

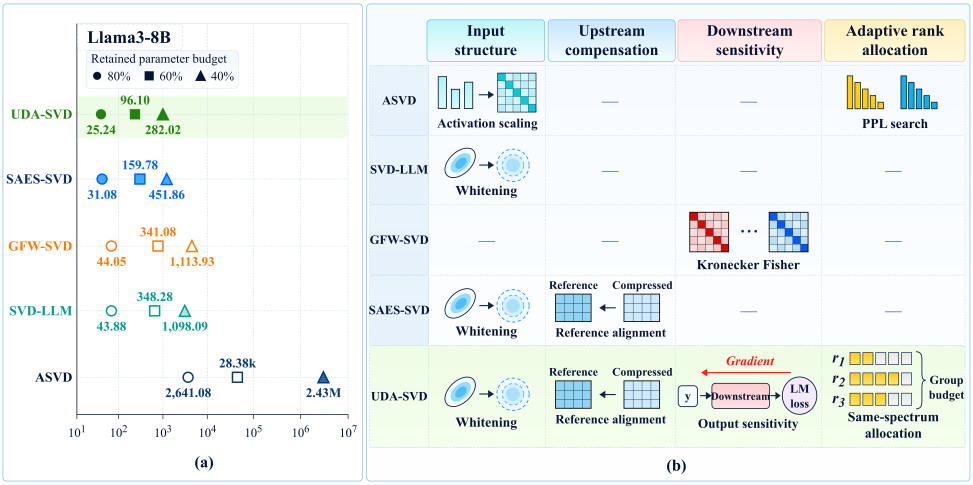


Figure 1: Left (a): WikiText-2 perplexity of LLaMA-3-8B on a logarithmic scale (lower is better). Right (b): Design mechanisms adopted by representative low-rank compression methods. Existing methods each cover only a subset of these mechanisms, with omitted ones marked by short horizontal lines. In contrast, UDA-SVD is the only method that addresses all three dimensions: upstream compensation, downstream sensitivity, and adaptive rank allocation.

jective to different degrees. Accordingly, sequential low-rank compression must account for two factors: upstream activation drift accumulated from previously compressed modules and the downstream sensitivity of the final objective to errors introduced at the current module.

However, existing methods generally account for at most one of these two effects. As shown in Fig. 1(b), ASVD (Yuan et al., 2023) rescales channels based on activation sensitivity, while SVD-LLM (Wang et al., 2024) whitens module inputs before truncation; both primarily improve module-wise reconstruction without explicitly modeling the upstream and downstream effects of sequential compression. Cumulative error-compensation methods, such as SAES-SVD (Hu et al., 2026), mitigate upstream activation drift by compensating for deviations inherited from previously compressed modules. By contrast, loss-aware methods, such as GFW-SVD (Chekalina et al., 2025), capture downstream error sensitivity by weighting module-level errors according to their estimated influence on the final loss. These two lines of work therefore address complementary aspects of sequential compression, but neither integrates upstream activation drift and downstream error sensitivity within a unified objective or uses the same objective to guide rank allocation. In addition, uniform rank allocation ignores differences in rank sensitivity across modules, resulting in inefficient use of a fixed parameter budget. Together, these limitations motivate a unified framework that incorporates both effects while allocating rank according to module sensitivity.

Accordingly, we propose **Upstream- and Downstream-Aware SVD (UDA-SVD)**, a post-training framework that incorporates upstream activation drift and downstream error sensitivity into both low-rank approximation and adaptive rank allocation. The core idea of UDA-SVD is to compress each module in the context of the partially compressed network: the approximation should compensate for deviations inherited from previously compressed modules while prioritizing error directions that have greater influence on the final language-modeling objective. Specifically, UDA-SVD consists of two closely connected components. ❶ **Upstream- and Downstream-Aware Low-Rank Approximation.** For each module, UDA-SVD collects paired activations from the full-precision and current compressed paths to characterize accumulated upstream activation drift, while second moments of output gradients serve as an empirical-Fisher-type proxy for downstream error sensitivity. These statistics are integrated into a unified quadratic objective that balances reconstruction on the current compressed path with alignment to the full-precision reference under a sensitivity-aware output metric. With fixed calibration statistics, the resulting rank-constrained problem reduces, through a two-sided transformation, to a standard low-rank approximation and therefore admits a closed-form solution via truncated SVD (Eckart & Young, 1936). ❷ **Objective-Consistent Adaptive Rank Allocation.** UDA-SVD further uses the singular-value spectrum of the same transformed matrix to quantify the marginal reduction in the unified objective from additional rank and adaptively allocate rank across modules under a fixed parameter budget. In this way, low-rank direction selection and rank allocation are no longer treated as independent decisions, but are driven by the same compression objective.

Extensive evaluations across multiple LLM families (Touvron et al., 2023a; Yang et al., 2025), model scales, compression budgets, and zero-shot benchmarks, as summarized in Fig. 1, show that UDA-SVD consistently delivers strong compression performance. These results demonstrate the effectiveness of jointly modeling upstream activation drift and downstream error sensitivity together with adaptive rank allocation for low-rank LLM compression. Our main contributions are threefold:

- We show that sequential low-rank compression is governed by both upstream activation drift and downstream error sensitivity, and formulate these two effects jointly rather than optimizing local reconstruction in isolation.
- We derive a unified quadratic objective that integrates reference-path compensation and downstream sensitivity. With fixed calibration statistics, the rank-constrained problem reduces through a two-sided transformation to standard low-rank approximation, yielding a closed-form truncated-SVD solution.
- We derive adaptive rank allocation from the same spectral objective used for low-rank approximation, unifying approximation and rank allocation under a fixed parameter budget. Extensive experiments validate the effectiveness of this design.

2 Related Work

Large Language Model Compression. LLM compression has been extensively studied to improve deployment efficiency while preserving model capability. Existing approaches mainly include pruning (Frantar & Alistarh, 2023; Ashkboos et al., 2024), quantization (Frantar et al., 2022; Lin et al., 2023; Hu et al., 2025), knowledge distillation (Hinton et al., 2014), and low-rank factorization (Yuan et al., 2023; Wang et al., 2024; Li et al., 2025). Pruning reduces model complexity by removing redundant weights or structures, but its practical efficiency often depends on the resulting sparsity pattern and hardware support. Quantization reduces numerical precision for substantial memory savings, but aggressive settings may degrade performance. Knowledge distillation transfers knowledge from a larger teacher to a compact student, typically requiring additional training. In contrast, low-rank factorization exploits redundancy in dense weights to obtain compact representations. SVD-based methods are particularly attractive for post-training compression because they preserve regular dense computation and provide flexible compression control through rank selection, while remaining compatible with adaptive and complementary compression techniques.

SVD-based Low-Rank Compression for LLMs. Within low-rank compression, truncated SVD provides a simple post-training solution, but directly minimizing weight-space approximation error ignores the data distribution and the network context in which each module operates. ASVD (Yuan et al., 2023) introduces activation-aware scaling to better preserve important activation channels, while SVD-LLM (Wang et al., 2024) applies input whitening to relate singular-value truncation to output reconstruction error. AdaSVD (Li et al., 2025) further compensates for truncation error and adapts compression ratios according to module importance. Beyond improving module-wise reconstruction, SAES-SVD (Hu et al., 2026) aligns compressed outputs with their full-precision references to suppress errors accumulated from previously compressed modules. GFW-SVD (Chekalina et al., 2025) incorporates Fisher-based sensitivity information to distinguish the importance of different approximation errors. Collectively, these methods extend SVD-based compression with activation awareness, accumulated-error compensation, sensitivity modeling, and adaptive compression. However, these components are generally introduced through separate objectives or auxiliary criteria rather than being integrated into a unified compression objective.

3 Preliminaries

We consider compressing a linear module l with weight $W_l \in \mathbb{R}^{m_l \times n_l}$ into a low-rank factorization $\widehat{W}_l = A_l B_l$, where $A_l \in \mathbb{R}^{m_l \times r_l}$ and $B_l \in \mathbb{R}^{r_l \times n_l}$. The parameter cost is $(m_l + n_l)r_l$. For a target retention ratio ρ, the ranks $\{r_l\}$ are constrained such that $\sum_l (m_l + n_l) r_l \approx \rho \sum_l m_l n_l$.

Activation-Aware Compression. Standard post-training SVD minimizes the reconstruction error on calibration inputs $X_l \in \mathbb{R}^{n_l \times N}$ from the current compressed path:

$$\min_{\mathrm{rank}(\widehat{W}_l) \le r_l} \|\widehat{W}_l X_l - W_l X_l\|_F^2. \tag{1}$$

Let $H_l = X_l X_l^\top$ be the input second moment. By defining R_l such that $H_l = R_l R_l^\top$, Eq. 1 is solved by truncating the SVD of $W_l R_l$. This adapts the approximation to the input geometry but ignores two critical factors in sequential compression.

Sequential Drift and Reference Alignment. In sequential compression, upstream replacements alter the inputs to module l. Let X_l^f and X_l denote inputs from the full-precision and current compressed paths, respectively, with drift $D_l = X_l^f - X_l$. The total output deviation from the full-precision reference $W_l X_l^f$ is:

$$\widehat{W}_l X_l - W_l X_l^f = \underbrace{(\widehat{W}_l - W_l) X_l}_{\text{new local error}} - \underbrace{W_l D_l}_{\text{inherited reference deviation}} \tag{2}$$

Standard SVD (Eq. 1) only minimizes the local error term $(\widehat{W}_l - W_l)X_l$, potentially preserving the inherited drift $-W_l D_l$. Methods like SAES-SVD (Hu et al., 2026) address this by adding a reference alignment term:

$$\|\widehat{W}_l X_l - W_l X_l\|_F^2 + \alpha_l \|\widehat{W}_l X_l - W_l X_l^f\|_F^2, \tag{3}$$

where α_l balances local reconstruction and reference alignment. However, Eq. 3 uses an unweighted Frobenius norm, treating all output directions equally. It fails to account for *downstream sensitivity*, i.e., that errors in certain output directions impact the final loss more severely. UDA-SVD addresses this by introducing a sensitivity-weighted metric into a unified objective that jointly handles drift compensation and directional error costs.

4 METHOD

We propose UDA-SVD, a unified post-training framework that addresses two critical mismatches in sequential low-rank compression: the *upstream activation drift* caused by preceding module replacements, and the heterogeneous *downstream error sensitivity* of output perturbations.

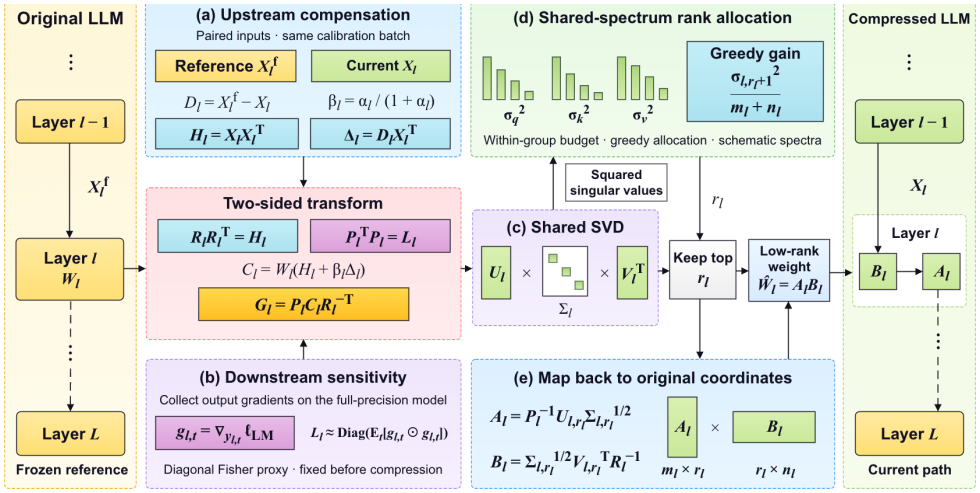


Figure 2: (a) Paired reference and compressed-path inputs yield the input second moment H_l and drift cross-statistic Δ_l. (b) Calibration-loss output gradients provide a diagonal sensitivity proxy L_l. Together with the compensation coefficient, these statistics define the two-sided transformed matrix G_l. (c–d) Its squared singular values determine marginal reconstruction benefits for rank allocation within each group budget. (e) The assigned rank controls truncation, after which the inverse transformation yields factors A_l and B_l with $\widehat{W}_l = A_l B_l$. After each group is replaced, the compressed-path inputs are refreshed; the full-precision reference and output metrics remain fixed. The diagram omits damping and ridge regularization for clarity; these are specified in Appendices A.3 and B.

4.1 A UNIFIED UPSTREAM- AND DOWNSTREAM-AWARE OBJECTIVE

Sequential compression introduces two coupled challenges. Replacing upstream modules perturbs the activations received by the current module, causing **upstream activation drift**. Consequently, matching the original weight on the current compressed path may preserve deviations from the full-precision reference. Meanwhile, errors introduced at the current module propagate through the remaining network, where their impact on the final loss depends on the **downstream error sensitivity** of specific output directions.

UDA-SVD addresses these by requiring each module to reconstruct an output that compensates for upstream drift while prioritizing directions sensitive to the final objective. We define a unified quadratic objective that measures residuals under a sensitivity-aware metric.

Weighted Compensation. Let $L_l \in \mathbb{R}^{m_l \times m_l}$ be a positive-semidefinite matrix representing downstream output sensitivity (estimated in Section 4.2). We measure reconstruction errors using the Mahalanobis-type norm induced by L_l:

$$\|E\|_{L_l}^2 := \mathrm{tr}(E^\top L_l E). \tag{4}$$

UDA-SVD applies this same metric to both local reconstruction on the compressed path and alignment with the full-precision reference:

$$\begin{aligned} \mathcal{J}_l(\widehat{W}_l) =& \|\widehat{W}_l X_l - W_l X_l\|_{L_l}^2 \\ &+ \alpha_l \|\widehat{W}_l X_l - W_l X_l^f\|_{L_l}^2, \end{aligned} \tag{5}$$

subject to $\mathrm{rank}(\widehat{W}_l) \le r_l$. Here, X_l denotes activations from the current compressed path, and X_l^f denotes activations from the full-precision reference path. The first term preserves consistency with the upstream compressed context, while the second term pulls the output toward the full-precision reference to compensate for accumulated drift. The scalar $\alpha_l \ge 0$ balances these two goals.

Let

$$\beta_l = \frac{\alpha_l}{1+\alpha_l}, \qquad D_l = X_l^f - X_l. \tag{6}$$

By completing the square, Eq. 5 can be rewritten as:

$$\mathcal{J}_l(\widehat{W}_l) = (1+\alpha_l)\|\widehat{W}_l X_l - Z_l\|_{L_l}^2 + k_l, \tag{7}$$

where $k_l = \alpha_l \|W_l D_l\|_{L_l}^2/(1+\alpha_l)$ is independent of $\widehat{W}_l$, and the compensated target Z_l is defined as:

$$Z_l = W_l(X_l + \beta_l D_l). \tag{8}$$

The coefficient $\beta_l \in [0,1)$ controls the degree of reference alignment. At $\beta_l = 0$ ($\alpha_l = 0$), the objective reduces to local reconstruction on the compressed path. As β_l increases, the target shifts toward the full-precision output, explicitly accounting for upstream activation drift without assuming it can be fully recovered by a rank-constrained map.

Sufficient Statistics. Let $Q \in \mathbb{R}^{m_l \times n_l}$ be the candidate weight matrix. The normalized objective, which preserves the rank-constrained minimizer, is:

$$\overline{\mathcal{J}}_l(Q) = \frac{\mathcal{J}_l(Q) - k_l}{1+\alpha_l} = \|QX_l - Z_l\|_{L_l}^2.$$

This objective depends on the data only through three sufficient statistics:

$$H_l = X_l X_l^\top, \qquad \Delta_l = D_l X_l^\top, \qquad C_l = Z_l X_l^\top. \tag{9}$$

Substituting Eq. 8 into C_l yields:

$$C_l = W_l(X_l + \beta_l D_l)X_l^\top = W_l(H_l + \beta_l \Delta_l). \tag{10}$$

Here, H_l captures the geometry of the current compressed input, while Δ_l captures the cross-correlation between upstream drift D_l and current inputs X_l. These statistics, along with the sensitivity metric L_l, fully characterize the local optimization problem.

4.2 Downstream Error Sensitivity

While reference alignment determines *what* output to recover, downstream sensitivity determines the *cost* of errors in different output directions. For a fixed context, let y denote the module output, $z = f_l(y)$ the subsequent logits, $p = \mathrm{softmax}(z)$ the predictive distribution, and $J_l = \partial z/\partial y$ the corresponding Jacobian. Under softmax cross-entropy, the Generalized Gauss-Newton (GGN) matrix is given by:

$$L_l^{\mathrm{GGN}} = J_l^\top \big(\mathrm{Diag}(p) - pp^\top\big) J_l \succeq 0. \tag{11}$$

This matrix equals the model Fisher information when the label expectation is taken under the predictive distribution and describes the second-order change in predictive KL divergence for small perturbations. It motivates a positive-semidefinite output metric, but generally differs from the full loss Hessian (Martens, 2020).

However, exact computation of Eq. 11 is intractable for large models. To obtain a tractable module-level proxy, we estimate L_l using the second moments of output gradients from the calibration loss, following the empirical Fisher approximation (Künstner et al., 2019). For a calibration dataset $\mathcal{T}$, let $g_{l,t} = \nabla_{y_{l,t}} \ell_{\mathrm{cal}}$ be the gradient of the language modeling loss with respect to the module output $y_{l,t}$. We compute:

$$\widehat{L}_l^{\mathrm{emp}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} g_{l,t} g_{l,t}^\top, \qquad g_{l,t} = \nabla_{y_{l,t}} \ell_{\mathrm{cal}}. \tag{12}$$

Here, $\mathcal{T}$ denotes the set of calibration output positions, and ℓ_{cal} is the language modeling loss for the corresponding sequence. This statistic measures the directional energy of gradients propagating from the loss, serving as an empirical-Fisher-type proxy for how much perturbations in specific output directions affect the final objective.

To ensure numerical stability and computational efficiency, we use a diagonal approximation with positive damping as our final metric L_l (see Appendices A.3, B and C for details on regularization and scaling). Crucially, because L_l is shared by both terms in Eq. 5, the upstream compensation and downstream sensitivity are integrated into a single weighted least-squares problem. With the objective, the metric, and the sufficient statistics all in place, the next subsection derives the closed-form rank-constrained solution.

4.3 Closed-Form Low-Rank Approximation via Two-Sided Transformation

With the unified objective and sensitivity metric defined, we derive the optimal low-rank approximation. Assuming $L_l, H_l \succ 0$, we minimize the normalized objective:

$$\min_{\mathrm{rank}(Q) \le r_l} \|L_l^{1/2}(QX_l - Z_l)\|_F^2. \tag{13}$$

We introduce invertible transformations based on the factorizations of the input and output metrics:

$$H_l = R_l R_l^\top, \qquad L_l = P_l^\top P_l. \tag{14}$$

Using $C_l = Z_l X_l^\top$, the unconstrained least-squares solution is

$$Q_l^\star = C_l H_l^{-1}. \tag{15}$$

Define the two-sided transformed matrix

$$\begin{aligned} G_l &= P_l Q_l^\star R_l \\ &= P_l C_l R_l^{-\top} \\ &= P_l W_l (H_l + \beta_l \Delta_l) R_l^{-\top}. \end{aligned} \tag{16}$$

Completing the square in Eq. 13 gives

$$\|L_l^{1/2}(QX_l - Z_l)\|_F^2 = \|P_l Q R_l - G_l\|_F^2 + \text{const}. \tag{17}$$

Since P_l and R_l are invertible, the variable substitution $Y = P_l Q R_l$ preserves the rank constraint. Thus, the problem reduces to a standard low-rank approximation:

$$\min_{\text{rank}(Y) \le r_l} \|Y - G_l\|_F^2. \tag{18}$$

Eq. 16 reveals how G_l integrates all three components: H_l (input geometry), $\beta_l \Delta_l$ (upstream drift compensation), and P_l (downstream sensitivity weighting).

Proposition 1 (Optimal Local Approximation). *Given fixed calibration statistics and positive-definite L_l, H_l, the global minimizer of Eq. 13 is obtained by truncating the SVD of G_l.*

Proof. By Eq. 18, the weighted objective reduces to a Frobenius norm minimization. By the Eckart–Young–Mirsky theorem, the optimal rank-r_l approximation of G_l is its top-r_l singular components, and the inverse transformation gives the factors of $\widehat{W}_l$. See Appendix B.4 for the full proof.

Specifically, let $G_l = U_l \Sigma_l V_l^\top$ be the SVD of G_l. The compressed weight $\widehat{W}_l = A_l B_l$ is reconstructed as:

$$A_l = P_l^{-1} U_{l,r_l} \Sigma_{l,r_l}^{1/2}, \qquad B_l = \Sigma_{l,r_l}^{1/2} V_{l,r_l}^\top R_l^{-1}. \tag{19}$$

This solution generalizes previous methods: setting $\beta_l = 0$ and $L_l = I$ recovers activation-aware reconstruction with input whitening, while setting only $L_l = I$ recovers reference-aligned SVD.

4.4 Adaptive Rank Allocation from Unified Spectrum

To determine the rank r_l for each module under a total parameter budget $\mathcal{B}$, we exploit the fact that the same transformed matrix G_l governs both direction selection and error magnitude. The squared singular values $\sigma_{l,j}^2$ of G_l represent the contribution of each component to the unified objective.

The excess error (tail energy) at rank r_l is $\mathcal{E}_l(r_l) = \sum_{j > r_l} \sigma_{l,j}^2$. Adding one more rank component reduces the objective by σ_{l,r_l+1}^2 at the cost of $m_l + n_l$ parameters. We define the **marginal objective reduction per parameter** as:

$$u_l(r_l + 1) = \frac{\sigma_{l,r_l+1}^2}{m_l + n_l}. \tag{20}$$

The rank allocation problem is formulated as:

$$\begin{aligned} &\min_{\{r_l\}_{l \in \mathcal{S}}} \sum_{l \in \mathcal{S}} \mathcal{E}_l(r_l), \\ &\text{s.t.} \quad \sum_{l \in \mathcal{S}} (m_l + n_l) r_l \le \mathcal{B}, \quad r_l^{\min} \le r_l \le r_l^{\max}. \end{aligned} \tag{21}$$

We solve this using a greedy strategy: starting from minimum ranks, we iteratively assign the next available rank to the module with the highest u_l. This ensures that the parameter budget is allocated to the directions that provide the greatest reduction in the unified upstream-downstream aware objective.

The entire process—statistics collection, transformation, SVD, and rank allocation—is performed sequentially in forward dependency order. After compressing a group of modules, the compressed path activations X_l are updated to reflect the upstream changes, ensuring that subsequent modules account for the accumulated drift. The specific workflow and algorithmic pseudocode for sequential compression are provided in Appendix F.

Table 1: Perplexity and zero-shot evaluation of LLaMA-3-8B across seven benchmark datasets under different target parameter-retention ratios. The table compares UDA-SVD with representative SVD-based compression methods. All results are averaged over seeds 7, 42, and 123. Lower perplexity and higher accuracy (%) are better. Boldface denotes the best mean at each budget and does not indicate statistical significance. Sample standard deviations are reported in Appendix E.1.

ρ	Method	Wiki2↓	PTB↓	C4↓	OBQA↑	ARC-E↑	WinoG.↑	HellaS.↑	ARC-C↑	PIQA↑	MathQ.↑	Avg.↑
	ASVD	2.43e+06	1.04e+06	1.35e+06	27.40	24.76	50.54	26.49	26.31	51.14	19.37	32.29
	SVD-LLM	1098.09	1999.16	738.22	26.27	34.33	49.41	26.99	22.04	55.22	22.27	33.79
0.4	GFW-SVD	1113.93	2009.37	746.99	26.60	34.65	49.99	27.11	22.04	55.35	22.51	34.04
	SAES-SVD	451.86	821.27	307.11	27.07	43.28	**53.75**	30.65	25.71	62.89	22.83	38.03
	UDA-SVD	**282.02**	**550.83**	**260.87**	**28.47**	**46.03**	53.41	**31.46**	**26.56**	**64.73**	**23.81**	**39.21**
	ASVD	28377.20	29966.34	78534.26	28.13	25.83	48.36	25.85	25.77	50.16	20.10	32.03
	SVD-LLM	348.28	645.66	273.08	31.87	51.91	56.56	37.03	31.83	66.72	24.39	42.90
0.6	GFW-SVD	341.08	635.99	271.62	31.80	51.68	56.51	36.83	31.20	66.45	24.32	42.69
	SAES-SVD	159.78	349.81	136.49	33.53	58.33	57.93	**44.14**	32.91	70.53	26.18	46.22
	UDA-SVD	**96.10**	**159.47**	**105.79**	**34.67**	**58.98**	**58.27**	43.29	**33.53**	**71.29**	**27.62**	**46.81**
	ASVD	2641.08	9725.05	3758.59	28.67	29.29	50.30	27.24	22.75	53.34	20.42	33.15
	SVD-LLM	43.88	93.87	52.21	38.13	66.76	68.19	59.43	41.41	75.08	29.50	54.07
0.8	GFW-SVD	44.05	93.50	52.21	38.47	66.84	**68.22**	59.32	41.78	75.08	29.49	54.17
	SAES-SVD	31.08	58.71	39.54	39.33	70.30	66.98	61.50	43.46	75.79	31.68	55.58
	UDA-SVD	**25.24**	**38.80**	**35.69**	**40.00**	**71.23**	67.22	**62.12**	**44.28**	**76.35**	**32.45**	**56.24**

5 EXPERIMENTS

We evaluate UDA-SVD on four LLMs across three compression budgets, focusing on: (1) overall compression quality against state-of-the-art baselines; (2) generalization across model families; and (3) the necessity of jointly modeling error metrics and rank allocation.

5.1 EXPERIMENTAL SETUP

Models & Baselines. We evaluate Qwen3 (0.6B, 8B) (Yang et al., 2025), LLaMA-3-8B (Grattafiori et al., 2024), and LLaMA-1-7B (Touvron et al., 2023a) at target retention ratios $\rho \in \{0.4, 0.6, 0.8\}$. Baselines include ASVD (Yuan et al., 2023), SVD-LLM (Wang et al., 2024), GFW-SVD (Chekalina et al., 2025), and SAES-SVD (Hu et al., 2026). All methods use identical calibration data and post-training pipelines without fine-tuning. See Appendix D for implementation details.

Metrics. We report perplexity on WikiText-2 (Merity et al., 2017), PTB (Marcus et al., 1993), and C4 (Raffel et al., 2020), and zero-shot accuracy on seven benchmarks (ARC (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), WinoGrande (Sakaguchi et al., 2020), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and MathQA (Amini et al., 2019)). *Average* denotes the unweighted mean accuracy. Task-level results and statistical details are in Appendix E.

5.2 OVERALL PERFORMANCE ON LLAMA-3-8B

Table 1 shows that UDA-SVD consistently outperforms all baselines on LLaMA-3-8B, achieving the lowest perplexity and highest average accuracy across all retention ratios. The simultaneous improvement in both language modeling and downstream tasks validates the effectiveness of integrating upstream drift compensation, downstream sensitivity, and adaptive rank allocation into a unified framework. While aggressive compression ($\rho = 0.4$) incurs inherent performance drops, UDA-SVD significantly mitigates this degradation compared to local reconstruction methods.

Beyond the accuracy metrics, low-rank approximation simultaneously decreases both computational complexity and parameter storage. This dual reduction substantially lowers the memory footprint and compute requirements. We evaluated the inference speed and memory usage of LLaMA-3-8B under different compression ratios on a single NVIDIA A6000 GPU. As illustrated in Figure 3, our proposed UDA-SVD achieves consistent acceleration over the FP16 baseline. As the retention ratio decreases from 80% to 40%, the memory footprint drops dramatically from the baseline 15.83GB to as low as 6.85GB. Meanwhile, the inference speedup ranges from 1.29× to 2.37×, demonstrating the practicality of UDA-SVD in accelerating inference while significantly reducing memory demands.

Table 2 compares UDA-SVD with the strong baseline SAES-SVD on other models. On Qwen3-8B, UDA-SVD maintains its superiority, confirming transferability to modern architectures. On the

smaller Qwen3-0.6B, gains are marginal but consistent, suggesting that even models with limited redundancy benefit from adaptive reallocation.

Notably, UDA-SVD also outperforms SAES-SVD on LLaMA-1-7B across all rank constraints (ρ). This consistent improvement highlights the robustness of our unified objective, demonstrating its effectiveness not only on newer models like Qwen3 but also on older architectures like LLaMA-1. The results suggest that the empirical Fisher proxy and the adaptive mechanism generalize well across diverse model structures. Full baseline comparisons are provided in Appendix E.2.

Table 2: Seven-task average accuracy of UDA-SVD and SAES-SVD on other models (%). All entries are means over seeds 7, 42, and 123. Boldface marks the best mean in each group.

Model	ρ **(Target parameter-retention ratio)**					
	0.4		**0.6**		**0.8**	
	SAES-SVD	**UDA-SVD**	SAES-SVD	**UDA-SVD**	SAES-SVD	**UDA-SVD**
Qwen3-8B	39.16	**39.21**	48.16	**48.86**	56.69	**58.03**
Qwen3-0.6B	34.20	**34.34**	36.31	**36.51**	39.10	**39.68**
LLaMA-1-7B	39.83	**41.52**	48.76	**49.78**	53.97	**55.04**

5.3 Ablation: Joint Error Metric and Allocation

To verify our core design principle—that rank allocation must be guided by the same unified error metric used for approximation—we ablate two components: (i) the *Unified Error Metric* (compensation + sensitivity) and (ii) *Adaptive Rank Allocation*. Results are in Table 3.

On LLaMA-3-8B, the full configuration yields the best performance. Crucially, enabling adaptive allocation *without* the unified metric provides no gain over the baseline. This confirms that spectral allocation is only effective when driven by an error signal that accurately captures both upstream and downstream effects. On Qwen3-8B, results are nuanced: while the full method wins at extreme budgets ($\rho = 0.4, 0.8$), simple adaptive allocation performs competitively at $\rho = 0.6$, suggesting that the empirical Fisher proxy may introduce slight estimation noise in intermediate regimes. Overall, the ablation supports the joint design, while indicating room for refining the sensitivity estimator.

Table 3: Ablation of joint compensation and weighting and dynamic budget allocation. ✓/× indicate enabled/disabled switches. Entries report seven-task average accuracy (mean ± sample standard deviation, %) across multiple random seeds.

Model	Comp. & weighting	Dyn. budget	$\rho = 0.4$	$\rho = 0.6$	$\rho = 0.8$
LLaMA-3-8B	×	×	33.51 ± 0.15	41.24 ± 0.08	53.21 ± 0.07
	✓	×	37.95 ± 0.18	44.93 ± 0.08	53.94 ± 0.21
	×	✓	33.39 ± 0.24	39.99 ± 0.10	51.25 ± 0.36
	✓	✓	**38.98 ± 0.36**	**47.02 ± 0.25**	**55.95 ± 0.04**
Qwen3-8B	×	×	36.94 ± 0.25	48.13 ± 0.69	56.49 ± 0.11
	✓	×	37.96 ± 0.62	46.89 ± 0.12	56.88 ± 0.45
	×	✓	38.50 ± 0.11	**49.08 ± 0.52**	57.58 ± 0.22
	✓	✓	**38.82 ± 0.48**	48.12 ± 0.58	**57.83 ± 0.17**

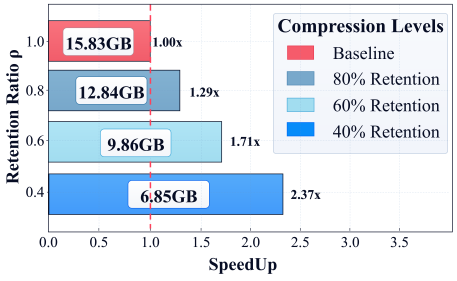


Figure 3: Memory usage and inference speedup of LLaMA-3-8B on varying retention ratios.

6 Conclusion

We presented UDA-SVD, a unified post-training framework for low-rank LLM compression that jointly models accumulated upstream activation drift and downstream error sensitivity. Through a two-sided transformation, its quadratic objective admits a closed-form truncated-SVD solution, while the resulting singular-value spectrum enables objective-consistent adaptive rank allocation under a fixed parameter budget. Experiments across four LLMs and three compression budgets show that UDA-SVD consistently achieves strong perplexity and zero-shot accuracy across model families and scales. Ablation studies verify the complementary benefits of unified error modeling and adaptive allocation, while deployment results demonstrate substantial memory savings and up to 2.37× inference speedup. Overall, UDA-SVD provides an accurate, efficient, and broadly applicable solution for low-rank LLM compression.

AI USE STATEMENT

The authors hereby declare the role of large language model (LLM) tools in the preparation of this manuscript: LLMs were solely utilized to assist with text polishing (including refining sentence structure, optimizing lexical expression, and enhancing language fluency) and **writing optimization** of the paper's narrative content.

It is explicitly emphasized that all core components of this research, which determine the originality, scientific validity, and academic value of the work, were independently completed by the research team through manual efforts. These components include, but are not limited to:

- The formulation and development of the overall research framework, core ideas, and logical structure of the study;
- The design, coding, debugging, and validation of all algorithms and program codes involved in the research;
- The design of experimental protocols, collection and preprocessing of experimental data, execution of experiments, analysis and interpretation of experimental results, and verification of conclusions.

The use of LLM tools did not involve any participation in the conception of research content, generation of technical solutions, implementation of experimental processes, or derivation of research conclusions. All content of this paper adheres to academic integrity standards, and the research team assumes full responsibility for the scientificity, authenticity, and originality of the work.

REPRODUCIBILITY STATEMENT

We provide detailed experimental settings to facilitate reproduction of our results. Appendix D specifies the evaluated models, compression budgets, calibration data, evaluation metrics, random seeds, and reporting conventions. The main comparisons and ablation studies are repeated across three random seeds, with means and sample standard deviations reported for aggregate accuracy and perplexity. Appendices A–C document the theoretical derivations, regularization, statistical scaling, and rank-allocation details; Appendix E provides task-level results; and Appendix F presents the complete sequential compression procedure. All compared methods use the same calibration data and evaluation pipeline. To support independent verification and future research, we will publicly release the source code, configuration files, data-processing and evaluation scripts, and commands required to reproduce the main experiments after publication.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2357–2367, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1227. URL https://aclanthology.org/P19-1227/.

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 7432–7439, 2020.

Viktoriia A. Chekalina, Daniil Moskovskiy, Tatyana Matveeva, Andrey Kuznetsov, and Evgeny Frolov. Generalized fisher-weighted svd: Scalable kronecker-factored fisher approximation for compressing large language models. May 2025. URL https://arxiv.org/abs/2505.17974.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332, 2022.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. *International Conference on Machine Learning*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems*, 2014.

Xing Hu, Yuan Cheng, Dawei Yang, Zukang Xu, Zhihang Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *arXiv preprint arXiv:2501.13987*, 2025.

Xing Hu, Dawei Yang, Yuan Cheng, Zhixuan Chen, and Zukang Xu. Saes-svd: Self-adaptive suppression of accumulated and local errors for svd-based llm compression. *arXiv preprint arXiv:2602.03051*, 2026.

Frederik Künstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Zhiteng Li, Mingyuan Xia, Jingyuan Zhang, Zheng Hui, Linghe Kong, Yulun Zhang, and Xiaokang Yang. Adasvd: Adaptive singular value decomposition for large language models. *arXiv preprint arXiv:2502.01403*, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research (JMLR)*, 2020.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, March 2024. URL https://arxiv.org/abs/2403.07378.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

A Output Curvature, Model Fisher, and Gradient Second Moments

A.1 The Full Hessian and the Generalized Gauss–Newton Matrix

For a general loss, a local perturbation of the module output satisfies

$$\ell(y+\delta y)-\ell(y)=\nabla_y\ell^\top\delta y+\tfrac{1}{2}\delta y^\top\nabla_y^2\ell\,\delta y+o(\|\delta y\|^2).$$

A directional quadratic form therefore describes only part of the loss change. In general, the first-order term cannot be omitted, nor can the quadratic form alone be identified with the actual loss difference. We distinguish the full Hessian, the positive-semidefinite GGN matrix, and the model Fisher below.

For a fixed context and label distribution q, let $z=f_l(y)$, $p=\mathrm{softmax}(z)$, and $\ell=-\sum_k q_k\log p_k$. Using $\partial p_i/\partial z_j=p_i(\mathbf{1}_{i=j}-p_j)$ gives

$$\nabla_z\ell=p-q,\qquad \nabla_z^2\ell=\mathrm{Diag}(p)-pp^\top, \tag{22}$$

where q is typically the one-hot vector of the observed label. For any vector v,

$$v^\top\big(\mathrm{Diag}(p)-pp^\top\big)v=\mathrm{Var}_{c\sim p}(v_c)\geq 0.$$

The logit-space curvature is thus positive semidefinite. The second-order chain rule yields

$$\nabla_y^2\ell=J_l^\top\big(\mathrm{Diag}(p)-pp^\top\big)J_l+\sum_k(p_k-q_k)\nabla_y^2 z_k. \tag{23}$$

The first term is the GGN matrix. The second accounts for second derivatives of the downstream network. The GGN equals the full Hessian only when the second term vanishes, for example when the logits are affine in y. Positive semidefiniteness makes the GGN suitable as a reconstruction metric; it does not imply that the full Hessian is positive semidefinite.

A.2 Definition and Interpretation of the Model Fisher

For a fixed context, treat y as the variable parameterizing the predictive distribution. Its model Fisher is

$$F_y=\mathbb{E}_{c\sim p(\cdot|y)}\left[\nabla_y\log p_c\,\nabla_y\log p_c^\top\right]. \tag{24}$$

Since $\nabla_z\log p_c=e_c-p$ and

$$\mathbb{E}_{c\sim p}[(e_c-p)(e_c-p)^\top]=\mathrm{Diag}(p)-pp^\top,$$

the chain rule gives

$$F_y=J_l^\top\big(\mathrm{Diag}(p)-pp^\top\big)J_l=L_l^{\mathrm{GGN}}. \tag{25}$$

The expectation is explicitly taken over labels sampled from the model's predictive distribution. This matrix also governs the local discrepancy between predictive distributions:

$$D_{\mathrm{KL}}(p(\cdot\mid y)\,\|\,p(\cdot\mid y+\delta y))=\tfrac{1}{2}\delta y^\top F_y\delta y+o(\|\delta y\|^2).$$

There is no first-order term at zero perturbation. This motivates a positive-semidefinite directional metric, but replacing F_y with empirical gradient statistics does not preserve the exact second-order KL term.

For language modeling losses over multiple prediction positions, the corresponding terms must be summed or averaged according to the loss convention. Vectorizing the module outputs of an entire sequence produces a curvature matrix with off-diagonal blocks coupling positions. Our objective instead applies the same $m_l \times m_l$ module metric at every output position and omits these inter-position couplings. It is consequently not an exact reformulation of full sequence curvature.

A.3 Empirical Gradient Statistics, Diagonal Approximation, and Damping

When L_l is only positive semidefinite, Eq. 4 defines a squared seminorm. The inverse transformation in the closed-form solution requires a positive-definite metric, which we obtain through damping.

Eq. 12 aggregates calibration-sequence gradients under observed labels across module output positions. Replacing the model-label expectation with observed labels has no general unbiasedness guarantee. Moreover, squaring a sequence-loss gradient introduces cross terms between contributions from different prediction positions. The result is a data-dependent gradient second-moment proxy, rather than the exact model Fisher, GGN, or Hessian. The local derivation requires only that the chosen metric be positive definite; it does not require these matrices to coincide.

For large models, we use

$$\widetilde{L}_l = \mathrm{Diag}\left(\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} g_{l,t}\odot g_{l,t}\right) + \mu_l I, \qquad \mu_l > 0. \tag{26}$$

Positive damping ensures $\widetilde{L}_l \succ 0$, so an invertible factor exists:

$$\widetilde{L}_l = \widetilde{P}_l^\top \widetilde{P}_l. \tag{27}$$

The diagonal approximation retains coordinatewise gradient energy and discards correlations between output coordinates. Damping adds an isotropic baseline to every direction. The metric requires $O(m_l)$ storage instead of $O(m_l^2)$, although input statistics and matrix decomposition incur additional costs. Appendix C.1 relates the mean and sum conventions for these statistics.

B Input Ridge Regularization and Numerical Stability

B.1 Ill-Conditioned Input Statistics

The input statistic $H_l = X_l X_l^\top$ can be singular or ill-conditioned when calibration data are limited, features are correlated, or some input channels are rarely active. Direct inversion can amplify estimation error and destabilize the input-side transformation. We therefore use the regularized objective

$$\mathcal{F}_l(Q) = \|\widetilde{L}_l^{1/2}(QX_l - Z_l)\|_F^2 + \lambda_l \|\widetilde{L}_l^{1/2} Q\|_F^2, \qquad \lambda_l > 0, \tag{28}$$

where $\widetilde{L}_l$ is the damped output metric in Eq. 26. The ridge term shrinks the weight map toward zero in the output-weighted norm and reduces sensitivity to poorly estimated input directions.

B.2 Normal Equations and the Regularized Transformation

Define

$$H_{l,\lambda} = H_l + \lambda_l I. \tag{29}$$

Expanding Eq. 28 and collecting terms that depend on Q gives

$$\mathcal{F}_l(Q) = \mathrm{tr}\Big[\widetilde{L}_l\big(QH_{l,\lambda}Q^\top - 2QC_l^\top\big)\Big] + \mathrm{const}. \tag{30}$$

Since $H_{l,\lambda} \succ 0$, the unconstrained optimum is

$$Q^{\star}_{l,\lambda} = C_l H^{-1}_{l,\lambda} = W_l(H_l + \beta_l \Delta_l)(H_l + \lambda_l I)^{-1}. \tag{31}$$

Let

$$\widetilde{L}_l = \widetilde{P}_l^{\top} \widetilde{P}_l, \qquad H_{l,\lambda} = R_{l,\lambda} R_{l,\lambda}^{\top}. \tag{32}$$

The regularized transformed matrix is then

$$\begin{aligned} G_{l,\lambda} &= \widetilde{P}_l Q^{\star}_{l,\lambda} R_{l,\lambda} \\ &= \widetilde{P}_l C_l R_{l,\lambda}^{-\top} \\ &= \widetilde{P}_l W_l (H_l + \beta_l \Delta_l) R_{l,\lambda}^{-\top}. \end{aligned} \tag{33}$$

Both output damping and input ridge retain the compensation statistic $\beta_l \Delta_l$. They nevertheless change the objective and its solution; they are not algebraically equivalent numerical operations on the unregularized problem. After truncating the SVD of $G_{l,\lambda}$, the factors are mapped back as in Eq. 19, using $\widetilde{P}_l$ and $R_{l,\lambda}$.

B.3 Completing the Square under the Rank Constraint

The regularized objective satisfies

$$\mathcal{F}_l(Q) = \|\widetilde{P}_l Q R_{l,\lambda} - G_{l,\lambda}\|_F^2 + \text{const}. \tag{34}$$

Because $\widetilde{P}_l$ and $R_{l,\lambda}$ are invertible, the transformation $Y = \widetilde{P}_l Q R_{l,\lambda}$ preserves rank. Thus the damped, ridge-regularized subproblem also reduces to

$$\min_{\text{rank}(Y) \le r_l} \|Y - G_{l,\lambda}\|_F^2, \tag{35}$$

and admits an optimal solution through exact truncated SVD. This guarantee concerns the regularized objective with fixed statistics.

B.4 Proof of Proposition 1

We prove the unregularized result stated in Proposition 1; the regularized counterpart follows by replacing L_l, H_l, P_l, R_l, and G_l with their damped and ridge-regularized versions in Appendix B.2. For clarity, define

$$F_l(Q) = \|P_l(QX_l - Z_l)\|_F^2,$$

where $L_l = P_l^{\top} P_l$, $H_l = X_l X_l^{\top} = R_l R_l^{\top}$, and $C_l = Z_l X_l^{\top}$. Because $L_l, H_l \succ 0$, both P_l and R_l are invertible. Expanding the objective gives

$$F_l(Q) = \text{tr}(P_l Q H_l Q^{\top} P_l^{\top}) - 2\,\text{tr}(P_l Q C_l^{\top} P_l^{\top}) + \|P_l Z_l\|_F^2.$$

The unconstrained least-squares solution is $Q_l^{\star} = C_l H_l^{-1}$. With $G_l = P_l Q_l^{\star} R_l$, we have

$$\|P_l Q R_l - G_l\|_F^2 = \text{tr}(P_l Q H_l Q^{\top} P_l^{\top}) - 2\,\text{tr}(P_l Q C_l^{\top} P_l^{\top}) + \|G_l\|_F^2,$$

where the cross term follows from $Q_l^{\star} H_l = C_l$. Therefore,

$$F_l(Q) = \|P_l Q R_l - G_l\|_F^2 + \|P_l Z_l\|_F^2 - \|G_l\|_F^2.$$

The final two terms are independent of Q. Hence minimizing Eq. 13 is equivalent to minimizing $\|P_l Q R_l - G_l\|_F^2$ under the same rank constraint.

Now let $Y = P_l Q R_l$. Since multiplication by invertible matrices preserves rank, $\mathrm{rank}(Y) = \mathrm{rank}(Q)$, and the mapping $Q \mapsto Y$ is bijective. The constrained problem is therefore exactly

$$\min_{\mathrm{rank}(Y)\le r_l} \|Y - G_l\|_F^2.$$

Let $G_l = U_l \Sigma_l V_l^\top$ and let $U_{l,r_l}\Sigma_{l,r_l}V_{l,r_l}^\top$ denote its rank-r_l truncated SVD. By the Eckart–Young–Mirsky theorem,

$$Y_l^\star = U_{l,r_l}\Sigma_{l,r_l}V_{l,r_l}^\top$$

is a global minimizer, with minimum approximation error $\sum_{j>r_l}\sigma_{l,j}^2$. Applying the inverse transformation yields

$$\widehat{W}_l = P_l^{-1} Y_l^\star R_l^{-1} = P_l^{-1} U_{l,r_l}\Sigma_{l,r_l}V_{l,r_l}^\top R_l^{-1}.$$

Writing $\widehat{W}_l = A_l B_l$ with the factors in Eq. 19 gives the claimed solution. If the truncation boundary contains repeated singular values, the minimizer need not be unique, but every corresponding rank-r_l truncated SVD remains globally optimal. □

C Statistical Scaling, Discrete Budgets, and Implementation

C.1 Objective Normalization and Statistical Scaling

The allocation problem in Eq. 21 minimizes the sum of the normalized objectives $\overline{\mathcal{J}}_l$, while Eq. 20 defines the marginal objective reduction per parameter used by the greedy allocator. If the original unnormalized objectives $\mathcal{J}_l$ were summed instead, the spectral tail for each module would be multiplied by $1 + \alpha_l$. When α_l varies across modules, this changes the allocation problem. Similarly, multiplying one module's output metric by a positive scalar leaves its fixed-rank optimum unchanged but rescales its squared spectrum and hence its marginal benefits relative to other modules. Statistical normalization is therefore part of the allocation criterion.

For clarity, the main text defines $H_l = X_l X_l^\top$ and $\Delta_l = D_l X_l^\top$. The implementation applies the same accumulation scale to both: for S fixed-length calibration sequences, it uses $2/S$ times the summed statistics. Provided this scale is shared within a group and the input ridge is scaled accordingly, it multiplies the group objective by a common positive constant and does not change the ordering of marginal benefits under a fixed budget.

Output statistics are obtained by backpropagating each sequence's mean next-token cross-entropy and summing squared gradients over all output positions. With $T = |\mathcal{T}|$, the implementation's sum statistic is $L_l^{\mathrm{sum}} = T\widehat{L}_l^{\mathrm{emp}}$. The corresponding damping scales satisfy

$$L_l^{\mathrm{sum}} + \mu_l^{\mathrm{sum}} I = T\left(\widehat{L}_l^{\mathrm{emp}} + \frac{\mu_l^{\mathrm{sum}}}{T} I\right). \tag{36}$$

Our diagonal implementation uses output damping $\mu_l^{\mathrm{sum}} = 10^{-6}$ under the sum convention. The initial input ridge is $\lambda_l = 10^{-4}\,\mathrm{tr}(H_l)/n_l$, with a positive floor and retries on factorization failure. Reproduction requires preserving the loss reduction, accumulation convention, and damping scale together. Changing the calibration length or sample count can change the relative influence of a fixed absolute output damping value.

C.2 Discrete Allocation and Its Optimality Scope

With statistics fixed, one additional rank reduces the local spectral error by

$$\mathcal{E}_l(r_l) - \mathcal{E}_l(r_l+1) = \sigma_{l,r_l+1}^2, \qquad u_l(r_l+1) = \frac{\sigma_{l,r_l+1}^2}{m_l + n_l}. \tag{37}$$

When all rank increments in a group have equal parameter cost, each module has nonincreasing marginal benefits, and repeatedly selecting the largest feasible benefit solves the corresponding separable integer allocation problem. In general, however, module dimensions differ and so do increment costs. Greedy selection by benefit-to-cost ratio is then not guaranteed to be globally optimal.

For example, with a remaining budget of 4, plain ratio-based greedy selection prefers an increment with cost 3 and benefit 5 over one with cost 4 and benefit 6. It leaves unusable budget and obtains a lower total benefit. We therefore treat the general allocation procedure as a discrete heuristic.

The minimum ranks must be feasible under the group budget before allocation begins. Every subsequent increment must respect both the maximum rank and remaining budget. The implementation additionally checks divisibility of remaining costs to reduce budget fragmentation. This check is neither sufficient for general feasibility nor a guarantee of global optimality. A group containing only one module has no opportunity for redistribution across modules under a fixed group budget.

C.3 Memory, Decomposition, and Factor Construction

For one module, the input statistics H_l and Δ_l each require $O(n_l^2)$ storage, a diagonal output metric requires $O(m_l)$, and explicitly forming the transformed matrix requires $O(m_l n_l)$. Blockwise accumulation avoids retaining all N paired activations during decomposition, but does not eliminate the cost of calibration forward and backward passes or reference-path caching. For a general dense matrix, full SVD costs $O(m_l n_l \min(m_l, n_l))$. Randomized low-rank decomposition may be used to approximate the required spectrum. In that case, the exact truncated-SVD result describes the ideal local optimum, while the computed solution also incurs numerical approximation error.

Factor orientations must be consistent with the inverse mapping. The input Cholesky factor satisfies $H = RR^\top$. If output Cholesky factorization is written as $L = CC^\top$, the transformation must use $P = C^\top$ so that $L = P^\top P$. A diagonal output metric can be applied through row scaling. In practice, inverse transformations should use linear solves, retaining symmetrization, precision promotion, and damping retries. Only H_l and the output metric require symmetrization; the cross-statistic Δ_l must not be forced to be symmetric.

Repeated singular values at the truncation boundary can make the optimal subspace nonunique. The use of the same spectrum in the main text means that allocation and factor construction correspond to the same weighted objective. The current implementation invokes spectral estimation and low-rank decomposition separately; it does not necessarily perform only one SVD per module. Once factors are obtained, a factorized implementation can evaluate $x \mapsto B_l x \mapsto A_l(B_l x)$. Appendix C.4 distinguishes this representation from the one used for quality evaluation.

C.4 Parameter Accounting and Evaluation Representation

Let $P_{\text{target}} = \sum_{l \in \mathcal{M}} m_l n_l$ denote the original parameter count of the target weights, P_{other} the count of all retained parameters outside those weights, and $P_{\text{factor}} = \sum_{l \in \mathcal{M}} (m_l + n_l) r_l$ the actual factor parameter count. The full-model parameter retention ratio under a factorized representation is

$$\rho_{\text{all}} = \frac{P_{\text{other}} + P_{\text{factor}}}{P_{\text{other}} + P_{\text{target}}} \approx \frac{P_{\text{other}} + \rho P_{\text{target}}}{P_{\text{other}} + P_{\text{target}}}.$$

Uncompressed embeddings and other retained parameters reduce the overall fraction of parameters removed relative to the target modules alone. Integer ranks further cause actual factor costs to differ from nominal budgets. We distinguish these quantities from the size of a serialized checkpoint.

Biases in target linear modules are retained and excluded from the decomposed-weight budget. For quality evaluation, the product $A_l B_l$ is written back into the original module as a dense weight, preserving the evaluation interface. Actual factorized deployment instead requires replacing the module with two consecutive linear operators. The reported budget therefore describes the parameter capacity of a factor representation, not the file size of the evaluated checkpoints. The factorized path in Figure 2 illustrates the representation defined by the method; the quality experiments use the dense-product evaluation procedure described here.

D Experimental Protocol

Models and compression settings. We evaluate UDA-SVD on Qwen3-0.6B, Qwen3-8B, LLaMA-3-8B, and LLaMA-1-7B at target linear-layer budget ratios $\rho \in \{0.4, 0.6, 0.8\}$. Appendix E

Table E.1: Detailed LLaMA-3-8B results corresponding to Table 1. Perplexity and Avg. report the mean ± sample standard deviation over seeds 7, 42, and 123; task columns report means. Boldface marks the best value at each budget.

ρ	Method	Wiki2 ↓	PTB ↓	C4 ↓	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
	ASVD	2.43e+06 ± 9.87e+05	1.04e+06 ± 3.50e+05	1.35e+06 ± 5.44e+05	27.40	24.76	26.31	50.54	26.49	51.14	19.37	32.29 ± 0.29
	SVD-LLM	1098.09 ± 8.91	1999.16 ± 51.61	738.22 ± 10.96	26.27	34.33	22.04	49.41	26.99	55.22	22.27	33.79 ± 0.39
0.4	GFW-SVD	1113.93 ± 8.70	2009.37 ± 36.07	746.99 ± 16.66	26.60	34.65	22.04	49.99	27.11	55.35	22.51	34.04 ± 0.26
	SAES-SVD	451.86 ± 29.36	821.27 ± 71.79	307.11 ± 23.04	27.07	43.28	25.71	**53.75**	30.65	62.89	22.83	38.03 ± 0.38
	UDA-SVD	**282.02 ± 3.87**	**550.83 ± 37.66**	**260.87 ± 6.64**	**28.47**	**46.03**	**26.56**	53.41	**31.46**	**64.73**	**23.81**	**39.21 ± 0.07**
	ASVD	28377.20 ± 1330.79	29966.34 ± 1209.84	78534.26 ± 2923.70	28.13	25.83	25.77	48.36	25.85	50.16	20.10	32.03 ± 0.08
	SVD-LLM	348.28 ± 7.45	645.66 ± 16.42	273.08 ± 9.41	31.87	51.91	31.83	56.56	37.03	66.72	24.39	42.90 ± 0.26
0.6	GFW-SVD	341.08 ± 5.83	635.99 ± 30.37	271.62 ± 7.50	31.80	51.68	31.20	56.51	36.83	66.45	24.32	42.69 ± 0.22
	SAES-SVD	159.78 ± 13.08	349.81 ± 15.02	136.49 ± 6.97	33.53	58.33	32.91	57.93	**44.14**	70.53	26.18	46.22 ± 0.09
	UDA-SVD	**96.10 ± 2.81**	**159.47 ± 4.27**	**105.79 ± 0.63**	**34.67**	**58.98**	**33.53**	**58.27**	43.29	**71.29**	**27.62**	**46.81 ± 0.12**
	ASVD	2641.08 ± 27.36	9725.05 ± 217.88	3758.59 ± 33.99	28.67	29.29	22.75	50.30	27.24	53.34	20.42	33.15 ± 0.20
	SVD-LLM	43.88 ± 0.90	93.87 ± 2.75	52.21 ± 1.33	38.13	66.76	41.41	68.19	59.43	75.08	29.50	54.07 ± 0.11
0.8	GFW-SVD	44.05 ± 0.82	93.50 ± 2.09	52.21 ± 1.28	38.47	66.84	41.78	**68.22**	59.32	75.08	29.49	54.17 ± 0.02
	SAES-SVD	31.08 ± 1.06	58.71 ± 3.21	39.54 ± 0.92	39.33	70.30	43.46	66.98	61.50	75.79	31.68	55.58 ± 0.20
	UDA-SVD	**25.24 ± 0.26**	**38.80 ± 1.86**	**35.69 ± 0.21**	**40.00**	**71.23**	**44.28**	67.22	**62.12**	**76.35**	**32.45**	**56.24 ± 0.32**

additionally reports LLaMA-2-13B results. All methods are applied post-training without fine-tuning. Compression statistics are estimated from 512 calibration sequences of length 256 sampled with the same stratified task-mixture protocol for all compared methods.

Evaluation. Perplexity is evaluated on WikiText-2, PTB, and C4 with sequence length 2048. Zero-shot accuracy is evaluated on OpenBookQA, ARC-Easy, ARC-Challenge, WinoGrande, HellaSwag, PIQA, and MathQA. WinoGrande uses `acc`, whereas the other six tasks use `acc_norm`. Avg. is the equally weighted arithmetic mean of the seven task metrics.

Seeds and reporting. Tables 1 and 2 use seeds 7, 42, and 123, whereas Table 3 uses seeds 42, 43, and 44. The paired groupwise comparison and the LLaMA-2-13B results in Appendix E use seed 42 and seeds 7, 42, and 123, respectively. Task-level columns report means. Avg. and perplexity report the mean and sample standard deviation over seeds when multiple seeds are available. Boldface marks the best unrounded mean within each comparison group and does not imply statistical significance.

Budget convention. The ratio ρ specifies the factor-parameter budget of the targeted linear layers. Biases and non-target parameters are retained. Consequently, ρ is neither a full-model retention ratio nor a serialized-checkpoint compression ratio. In the paired groupwise comparison, uniform and dynamic allocation use matched group budgets; small differences in the realized parameter count arise only from integer rank allocation.

E Detailed Experimental Results

This appendix expands the aggregate results in the main text with task-level measurements and an additional evaluation on LLaMA-2-13B. To keep the presentation compact, results that share an experimental question are grouped as panels within a single table. OBQA, ARC-E, ARC-C, WinoG., and HellaS. denote OpenBookQA, ARC-Easy, ARC-Challenge, WinoGrande, and HellaSwag, respectively. WinoGrande uses `acc`; the remaining six tasks use `acc_norm`. All accuracy values are percentages. Task columns report means, whereas Avg. and perplexity entries report the mean and sample standard deviation over the stated seeds.

E.1 Overall Results on LLaMA-3-8B

Table E.1 expands the LLaMA-3-8B comparison in Table 1. UDA-SVD achieves the best average accuracy and the lowest perplexity on all three corpora at each compression budget. The task-level results also show that the aggregate improvement does not require winning every individual benchmark: SAES-SVD or GFW-SVD remains competitive on isolated tasks, while UDA-SVD provides the strongest overall balance.

E.2 Cross-Model Task Results

Table E.2 groups the two Qwen3 models and LLaMA-1-7B, which are summarized in Table 2. Relative to SAES-SVD, UDA-SVD improves the average accuracy at every reported budget for all three models. The gains are smaller on Qwen3-0.6B than on the two larger models, consistent with the reduced freedom for reallocating rank in the smaller network.

E.3 Component Ablations

Table E.3 reports the task-level measurements behind Table 3. On LLaMA-3-8B, jointly enabling compensation, output weighting, and dynamic allocation gives the highest average accuracy at every budget. On Qwen3-8B, the complete configuration is best at $\rho = 0.4$ and $\rho = 0.8$, whereas dynamic allocation without compensation and weighting is best at $\rho = 0.6$. This exception supports interpreting the components jointly while retaining the model- and budget-specific variation visible in the data.

E.4 Controlled Groupwise Allocation

Table E.4 isolates the allocation rule by matching compensation, output weighting, and group budgets. Replacing uniform ranks with spectral allocation raises average accuracy from 46.88 to 48.30 on Qwen3-8B and from 44.89 to 46.27 on LLaMA-3-8B. These gains are obtained under matched group budgets, although individual task scores can remain unchanged or decrease slightly.

E.5 Additional Results on LLaMA-2-13B

Table E.5 reports both accuracy and perplexity for LLaMA-2-13B. UDA-SVD achieves the highest average accuracy and the lowest mean perplexity at $\rho = 0.4$ and $\rho = 0.6$. At $\rho = 0.8$, UDA-SVD matches SVD-LLM to the reported precision for the highest average accuracy (57.35%) and achieves the lowest mean perplexity (37.58, compared with 37.59 for SVD-LLM). These results show that the gains are largest under stronger compression and narrow at the highest retention ratio.

Table E.2: Task-level accuracy on Qwen3-8B, Qwen3-0.6B, and LLaMA-1-7B corresponding to Table 2. Task columns report means over seeds 7, 42, and 123; Avg. reports the mean ± sample standard deviation of the per-seed seven-task averages. Boldface marks the best value at each budget.

(a) Qwen3-8B

ρ	Method	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
0.4	ASVD	26.87	26.59	25.91	51.07	26.36	52.09	18.79	32.52 ± 0.36
	SVD-LLM	**28.20**	38.87	25.97	53.51	31.78	60.45	24.65	37.63 ± 0.12
	GFW-SVD	28.07	38.95	25.88	53.38	31.68	60.59	24.54	37.58 ± 0.19
	SAES-SVD	27.67	43.00	26.34	**53.72**	**32.82**	**64.47**	26.13	39.16 ± 0.45
	UDA-SVD	28.07	**43.32**	**26.73**	53.70	31.90	64.13	**26.62**	**39.21 ± 0.16**
0.6	ASVD	28.33	30.85	26.76	50.46	31.44	56.00	19.75	34.80 ± 0.10
	SVD-LLM	**35.00**	56.68	**38.31**	61.09	47.90	70.33	30.61	48.56 ± 0.06
	GFW-SVD	**35.00**	56.96	38.28	**61.83**	**47.93**	70.11	30.90	48.72 ± 0.25
	SAES-SVD	33.73	60.16	36.15	59.67	45.11	**70.37**	31.94	48.16 ± 0.55
	UDA-SVD	34.07	**62.01**	37.66	59.06	44.69	69.88	**34.64**	**48.86 ± 0.31**
0.8	ASVD	31.33	43.76	32.03	57.33	58.04	67.14	23.29	44.70 ± 0.22
	SVD-LLM	**40.07**	69.78	47.95	65.93	64.80	**74.76**	37.43	57.25 ± 0.63
	GFW-SVD	39.73	69.60	47.41	66.19	**64.84**	74.63	37.39	57.11 ± 0.47
	SAES-SVD	39.00	70.95	46.27	65.82	60.42	73.78	40.58	56.69 ± 0.72
	UDA-SVD	38.07	**74.79**	**49.09**	**66.35**	60.44	73.94	**43.53**	**58.03 ± 0.17**

(b) Qwen3-0.6B

ρ	Method	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
0.4	ASVD	27.07	25.81	**27.13**	50.38	26.23	50.29	19.63	32.36 ± 0.36
	SVD-LLM	25.33	27.61	23.49	50.75	26.88	52.45	22.51	32.72 ± 0.02
	GFW-SVD	25.33	27.78	22.90	50.49	26.98	52.21	22.09	32.54 ± 0.03
	SAES-SVD	24.80	31.69	22.53	**52.38**	27.72	**56.64**	**23.66**	34.20 ± 0.17
	UDA-SVD	**27.67**	**32.20**	22.01	51.59	**27.75**	56.15	23.00	**34.34 ± 0.23**
0.6	ASVD	**29.73**	24.38	**25.80**	50.36	26.28	51.43	18.46	32.35 ± 0.21
	SVD-LLM	27.07	31.73	24.18	50.96	29.69	56.86	22.75	34.75 ± 0.16
	GFW-SVD	27.27	31.58	24.46	50.72	29.68	56.86	22.64	34.74 ± 0.18
	SAES-SVD	26.40	37.37	24.63	**53.04**	29.19	58.67	**24.87**	36.31 ± 0.04
	UDA-SVD	27.20	**38.02**	23.95	52.59	**30.30**	**59.01**	24.49	**36.51 ± 0.10**
0.8	ASVD	26.67	25.56	27.13	49.36	26.24	51.96	19.88	32.40 ± 0.31
	SVD-LLM	**30.80**	42.02	27.33	53.46	**36.23**	61.43	25.76	39.58 ± 0.24
	GFW-SVD	**30.80**	41.37	**27.65**	52.49	35.98	61.37	25.53	39.31 ± 0.09
	SAES-SVD	27.80	44.12	26.73	53.78	33.28	61.35	**26.63**	39.10 ± 0.39
	UDA-SVD	29.20	**45.82**	26.25	**53.88**	35.05	**61.99**	25.56	**39.68 ± 0.29**

(c) LLaMA-1-7B

ρ	Method	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
0.4	ASVD	24.40	27.09	27.67	50.36	26.41	49.58	21.68	32.46 ± 0.35
	SVD-LLM	28.20	39.90	25.34	51.17	31.33	59.39	20.82	36.59 ± 0.24
	GFW-SVD	28.20	39.98	25.34	51.64	31.33	59.38	20.75	36.66 ± 0.21
	SAES-SVD	30.00	45.76	27.05	53.35	34.50	64.07	22.65	39.83 ± 0.13
	UDA-SVD	**31.20**	**49.33**	**27.90**	**54.78**	**37.07**	**67.16**	**23.23**	**41.52 ± 0.34**
0.6	ASVD	25.60	28.24	26.28	49.70	27.40	52.87	20.32	32.92 ± 0.16
	SVD-LLM	36.47	56.48	32.65	61.40	49.77	70.60	23.02	47.20 ± 0.48
	GFW-SVD	36.73	56.75	32.74	61.40	49.89	70.66	23.24	47.34 ± 0.38
	SAES-SVD	36.73	59.05	34.36	59.06	49.41	71.71	23.96	48.76 ± 0.25
	UDA-SVD	**38.13**	**61.71**	**35.38**	**61.64**	**53.67**	**73.16**	**24.79**	**49.78 ± 0.28**
0.8	ASVD	33.40	44.19	29.24	61.06	48.91	65.00	21.13	43.28 ± 0.16
	SVD-LLM	**43.20**	65.29	38.25	68.09	**66.47**	76.48	25.05	54.69 ± 0.29
	GFW-SVD	43.07	65.49	37.94	**68.17**	66.40	76.42	24.90	54.63 ± 0.31
	SAES-SVD	41.47	66.57	38.57	65.93	63.67	**76.59**	24.99	53.97 ± 0.21
	UDA-SVD	42.20	**67.79**	**39.36**	67.32	66.14	76.53	**25.92**	**55.04 ± 0.21**

Table E.3: Task-level ablation results corresponding to Table 3. C&W denotes compensation and output weighting; Dyn. denotes dynamic rank allocation. Task columns report means over seeds 42, 43, and 44, while Avg. reproduces the main-text mean ± sample standard deviation.

(a) LLaMA-3-8B

ρ	C&W	Dyn.	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
0.4	×	×	25.60	33.38	21.62	49.75	26.77	54.91	22.57	33.51 ± 0.15
	✓	×	27.40	42.94	25.43	**53.20**	30.59	63.18	22.88	37.95 ± 0.18
	×	✓	26.27	32.34	21.22	49.99	26.77	54.82	22.30	33.39 ± 0.24
	✓	✓	**29.40**	**45.16**	**26.37**	53.12	**31.28**	**64.15**	**23.42**	**38.98 ± 0.36**
0.6	×	×	30.40	49.65	29.32	53.78	35.72	65.32	24.51	41.24 ± 0.08
	✓	×	33.40	56.30	31.63	55.75	40.88	70.08	26.45	44.93 ± 0.08
	×	✓	28.60	47.62	27.87	53.33	33.53	64.29	24.67	39.99 ± 0.10
	✓	✓	**35.13**	**60.00**	**33.67**	**58.25**	**43.25**	**71.42**	**27.40**	**47.02 ± 0.25**
0.8	×	×	38.20	65.91	41.21	66.30	57.43	74.48	28.94	53.21 ± 0.07
	✓	×	37.67	68.03	41.24	65.04	58.89	75.44	31.30	53.94 ± 0.21
	×	✓	37.47	62.99	38.20	64.80	53.25	73.01	29.00	51.25 ± 0.36
	✓	✓	**39.13**	**70.34**	**43.77**	**67.53**	**62.14**	**76.39**	**32.34**	**55.95 ± 0.04**

(b) Qwen3-8B

ρ	C&W	Dyn.	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
0.4	×	×	27.67	37.85	25.17	52.75	30.94	60.10	24.09	36.94 ± 0.25
	✓	×	27.20	40.29	25.14	52.67	31.39	63.08	25.96	37.96 ± 0.62
	×	✓	**29.80**	41.05	**26.93**	**53.83**	31.88	61.35	24.66	38.50 ± 0.11
	✓	✓	28.27	**42.63**	25.37	53.06	**31.97**	**63.78**	**26.63**	**38.82 ± 0.48**
0.6	×	×	34.67	56.55	37.26	**61.75**	45.85	70.09	30.75	48.13 ± 0.69
	✓	×	32.40	58.04	33.99	58.12	42.51	69.50	33.67	46.89 ± 0.12
	×	✓	**35.67**	57.25	**38.11**	61.56	**48.50**	**70.98**	31.49	**49.08 ± 0.52**
	✓	✓	33.93	**59.47**	36.01	58.70	44.33	70.02	**34.36**	48.12 ± 0.58
0.8	×	×	**39.87**	67.97	47.61	65.30	63.32	74.41	36.97	56.49 ± 0.11
	✓	×	39.27	71.83	47.67	65.38	58.32	73.54	42.17	56.88 ± 0.45
	×	✓	39.60	70.79	48.83	**66.35**	**64.87**	**75.01**	37.62	57.58 ± 0.22
	✓	✓	39.07	**74.06**	**48.92**	65.38	60.19	74.01	**43.19**	**57.83 ± 0.17**

Table E.4: Task-level accuracy for paired groupwise allocation configurations (%). Both configurations use seed 42 and $\rho = 0.6$, with identical compensation, output weighting, and group budgets.

Model	Configuration	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
Qwen3-8B	Uniform	31.20	58.84	34.98	57.06	42.19	69.10	34.81	46.88
	UDA-SVD	**32.00**	**60.40**	**37.54**	**59.19**	**43.86**	**69.97**	**35.18**	**48.30**
LLaMA-3-8B	Uniform	33.20	57.07	31.14	55.41	40.71	**70.40**	**26.30**	44.89
	UDA-SVD	**34.40**	**58.96**	**31.83**	**59.59**	**42.47**	**70.40**	26.26	**46.27**

Table E.5: Detailed accuracy and perplexity results on LLaMA-2-13B. Task-level accuracy columns report means over seeds 7, 42, and 123, whereas Avg. and all perplexity entries report the mean ± sample standard deviation across the same seeds. Higher accuracy and lower perplexity are better; the best entries within each budget are boldfaced.

(a) Task-level accuracy (%)

ρ	Method	OBQA	ARC-E	ARC-C	WinoG.	HellaS.	PIQA	MathQA	Avg. ↑
0.4	ASVD	24.87	26.37	26.19	50.30	26.22	49.24	20.48	31.95 ± 0.04
	SVD-LLM	28.00	41.57	23.63	52.17	29.74	60.36	23.56	37.00 ± 0.16
	GFW-SVD	28.20	41.43	23.75	51.85	29.72	60.25	23.55	36.96 ± 0.16
	SAES-SVD	30.73	49.99	27.87	54.14	34.77	**65.92**	**24.65**	41.15 ± 0.37
	UDA-SVD	**32.20**	**50.59**	**29.10**	**54.49**	**35.01**	65.65	24.29	**41.62 ± 0.31**
0.6	ASVD	28.40	36.49	24.91	56.96	34.52	59.01	21.60	37.41 ± 0.05
	SVD-LLM	36.93	62.33	35.41	62.14	46.70	72.34	25.66	48.79 ± 0.26
	GFW-SVD	36.67	62.39	35.61	62.27	46.78	72.34	25.63	48.81 ± 0.27
	SAES-SVD	**38.40**	63.47	35.72	62.33	49.75	72.52	**27.00**	49.89 ± 0.10
	UDA-SVD	**38.40**	**65.01**	**37.63**	**63.27**	**51.50**	**72.56**	26.73	**50.73 ± 0.14**
0.8	ASVD	39.67	60.73	36.26	65.54	59.27	73.41	24.95	51.40 ± 0.07
	SVD-LLM	42.27	72.21	**43.80**	**71.30**	**66.14**	77.08	28.64	**57.35 ± 0.25**
	GFW-SVD	42.00	**72.59**	43.00	70.40	65.49	77.04	28.87	57.06 ± 0.20
	SAES-SVD	42.20	72.21	42.80	69.88	64.55	**77.22**	**29.40**	56.89 ± 0.15
	UDA-SVD	**42.40**	72.11	43.74	71.24	66.12	77.15	28.70	**57.35 ± 0.28**

(b) Perplexity

ρ	Method	WikiText-2 ↓	PTB ↓	C4 ↓	Mean ↓
0.4	ASVD	5453.25 ± 467.81	3626.23 ± 359.44	3950.09 ± 106.05	4343.19 ± 242.76
	SVD-LLM	147.22 ± 8.30	773.73 ± 18.25	174.94 ± 5.85	365.30 ± 10.19
	GFW-SVD	146.83 ± 8.09	770.72 ± 18.33	174.49 ± 6.21	364.01 ± 9.72
	SAES-SVD	90.67 ± 4.25	594.36 ± 28.61	104.71 ± 2.35	263.25 ± 11.29
	UDA-SVD	**83.40 ± 3.29**	**559.81 ± 26.74**	**99.67 ± 2.95**	**247.63 ± 6.85**
0.6	ASVD	183.99 ± 2.50	1761.64 ± 19.80	228.09 ± 3.22	724.58 ± 7.49
	SVD-LLM	31.21 ± 0.13	262.87 ± 3.13	42.61 ± 0.63	112.23 ± 1.18
	GFW-SVD	31.17 ± 0.11	262.18 ± 2.36	42.55 ± 0.62	111.97 ± 0.99
	SAES-SVD	31.25 ± 0.53	272.31 ± 5.82	40.52 ± 0.47	114.69 ± 1.67
	UDA-SVD	**24.53 ± 0.22**	**209.68 ± 8.38**	**33.64 ± 0.26**	**89.28 ± 2.72**
0.8	ASVD	13.80 ± 0.00	255.44 ± 1.00	16.95 ± 0.00	95.40 ± 0.33
	SVD-LLM	10.15 ± 0.06	87.84 ± 2.63	14.76 ± 0.04	37.59 ± 0.89
	GFW-SVD	10.47 ± 0.02	91.92 ± 1.03	14.76 ± 0.12	39.05 ± 0.35
	SAES-SVD	12.05 ± 0.11	106.63 ± 2.07	17.00 ± 0.07	45.23 ± 0.69
	UDA-SVD	**10.14 ± 0.07**	**87.84 ± 2.41**	**14.75 ± 0.06**	**37.58 ± 0.81**

F SEQUENTIAL COMPRESSION PROCEDURE

UDA-SVD compresses target modules group by group in forward-dependency order:

$$\{q, k, v\} \longrightarrow \{o\} \longrightarrow \{\text{up}, \text{gate}\} \longrightarrow \{\text{down}\}.$$

Modules within a group share the current compressed-path state when their input statistics are collected. The full-precision reference model and damped output metrics remain fixed, whereas compressed-path inputs are refreshed after every group replacement. Each group budget is initialized from the prescribed factor-parameter budget, and capacity is redistributed only within that group.

Algorithm 1: Sequential UDA-SVD compression

Input: Full-precision model M_f; calibration data $\mathcal{D}$; compensation coefficients $\{\alpha_l\}$; group budgets $\{\mathcal{B}_\mathcal{S}\}$; feasible rank bounds $\{r_l^{\min}, r_l^{\max}\}$.
Output: Low-rank factors $\{A_l \in \mathbb{R}^{m_l \times r_l},\ B_l \in \mathbb{R}^{r_l \times n_l}\}$ for all target modules.

$M_c \leftarrow \text{Copy}(M_f)$
$\{L_l\} \leftarrow \text{EstimateDiagonalOutputMetric}(M_f, \mathcal{D})$ with positive damping
Keep M_f and $\{L_l\}$ fixed throughout compression
foreach *Transformer block b in forward order* **do**
  foreach *group $\mathcal{S}$ in* $\{q, k, v\} \to \{o\} \to \{\text{up}, \text{gate}\} \to \{\text{down}\}$ **do**
    $\{(X_l^f, X_l)\}_{l \in \mathcal{S}} \leftarrow \text{CollectPairedInputs}(M_f, M_c, \mathcal{D}, b, \mathcal{S})$
    foreach *module $l \in \mathcal{S}$* **do**
      $\beta_l \leftarrow \alpha_l / (1 + \alpha_l)$
      $H_l \leftarrow X_l X_l^\top$
      $\Delta_l \leftarrow (X_l^f - X_l) X_l^\top$
      $(G_l, P_l, R_l) \leftarrow \text{BuildTransforms}(W_l, H_l, \Delta_l, L_l, \beta_l)$
      $\boldsymbol{\sigma}_l \leftarrow \text{SingularValues}(G_l)$
    $\{r_l\}_{l \in \mathcal{S}} \leftarrow \text{SpectralRankAllocation}\big(\{\boldsymbol{\sigma}_l\}, \{m_l + n_l\}, \mathcal{B}_\mathcal{S}, \{r_l^{\min}, r_l^{\max}\}\big)$
    foreach *module $l \in \mathcal{S}$* **do**
      $(U_{l,r_l}, \Sigma_{l,r_l}, V_{l,r_l}) \leftarrow \text{TruncatedSVD}(G_l, r_l)$
      $A_l \leftarrow P_l^{-1} U_{l,r_l} \Sigma_{l,r_l}^{1/2}$
      $B_l \leftarrow \Sigma_{l,r_l}^{1/2} V_{l,r_l}^\top R_l^{-1}$
    `// Replace the group only after all factors in the group are computed.`
    $M_c \leftarrow \text{ReplaceGroup}(M_c, b, \mathcal{S}, \{A_l B_l\}_{l \in \mathcal{S}})$
    $\text{RefreshCompressedPath}(M_c, \mathcal{D}, b, \mathcal{S})$
return $\{A_l, B_l\}$

BUILDTRANSFORMS implements Eqs. 14–16, using the damped output metric in Appendix A.3 and the input regularization in Appendix B when required. SPECTRALRANKALLOCATION solves Eq. 21 with the marginal utility in Eq. 20, subject to the feasibility conditions in Appendix C.2. The factors follow the inverse mapping in Eq. 19. This procedure is a sequential realization of the fixed-statistics local solver and does not claim joint global optimality for the full network.